- ❖ All resources related to the project will be shared here
- ❖ Every weekly resource will also contain a weekly task, which is expected to be completed before the next week's resources are updated
- ❖ We would highly appreciate it if you keep following these weekly tasks on time since it would avoid a lot of last-minute work for you guys, but at the same time, if any of you are not able to cover it up in the given time due to whatever reason then let us know about it and try covering it up later
- ❖ We want it to be a learning experience for you all and not some mandatory assignment with a hard deadline, so learn at your pace and make the best out of this opportunity

Checkpoint-1: EDA + Text Pre-Processing

For this week, you will be brushing up on your Exploratory Data Analysis basics and also receiving a basic introduction to Text Pre-Processing.

➢ Exploratory Data Analysis

Watch the videos numbered 6-12 & 14 in this playlist

Feel free to skip any topic you are already confident in

➢ Text Pre-Processing

1. Read articles 1 & 2
2. For Python implementation, refer to videos numbered 2-10 in this playlist

Assignment-1:

It's time to implement what you have studied through the above-mentioned resource.
We will be using this dataset
The csv file has the following columns -

- ItemID : Unique ID for tweets
- Sentiment : 1 if tweet is of depression sentiment, 0 if not
- SentimentSource : Source of tweet extraction
- SentimentText : Raw text of tweet

Task 1 - Introduction to Colab

Go through an introduction on google colab and implement the following:

- Mount your drive in colab
- Upload the csv file in a Google Drive folder
- Import it in colab as a DataFrame (df)
  Note: Use pd.read_csv("path of the file",error_bad_lines=False)
- Preview top 5 rows of df

Task 2 - Text Pre Processing

-
- Drop unnecessary columns (ID, Source)
- Remove punctuations like . , ! $( ) * % @
- Remove URLs
- Perform lower casing
- Perform tokenization
- Remove Stopwords
- Perform stemming
- Perform lemmatization

Task 3 - Exploratory Data Analysis

- Make word cloud from the overall dataset
- Make word clouds for individual classes (Positive / Negative)
- Get sentence lengths (word count) for both classes' data and create plots to examine whether there is a difference in tweet lengths between depressive and non-depressive content
- Get counts of tweets for both classes and make a count plot
- Check the longest tweet in the dataset for both classes

Task 4 - Setting up Github Repository

Create a github repository and store your Python notebooks in it
Feel free to refer to this article for an introduction to github and this article for uploading your colab notebooks to your github repository.

Checkpoint-2: LSTM+ Word2Vec

This week you will learn about the basics of Machine Learning and a popular machine learning algorithm called Long Short Term Memory (LSTM). You will also learn about Vector Conversion of Words(Word2Vec) and the most exciting part, Scraping Twitter Data(Optional).
Feel free to skip any topic you are already familiar with.

➢ Scraping Twitter Data

Use Textblob to classify tweets as positive or negative.
You can follow this video tutorial and article while seeking help with the code
However, you can ignore the scraping part for now if you can't do it since it is not mandatory

➢ Basic ML (Skip if you already know)

Introduction to ML- Video
Linear regression - 1, 2
Logistic Regression - 1

➢ LSTM

Basics- 1, 2

Further, learn about this simple LSTM model for Sentiment Analysis and understand how it incorporates sequence for final prediction in contrast to looking at individual words as conventional ML models do

➢ Word2Vec

We aim to transform words into numerical representations for the machine to understand. To achieve this, we'll convert them into vectors using the word2vec technique
Gain an understanding from this video and article

Assignment-2:

It's time to implement what you have studied through the above-mentioned resources. This week we will implement an LSTM model, perform word-to-vector conversion and then compare our model to logistic regression.
We will be using the same dataset this week.
Use test_size=0.3 for splitting the data

Task 1-
Implement a simple LSTM for our dataset and determine the model's test accuracy

Task 2-

Incorporate Word2Vec embeddings into our dataset & integrate them with the LSTM model and determine the model's test accuracy
Help

Task 3-
Apply logistic regression to our dataset and compare it's performance with our model
Plot the loss vs epoch and accuracy vs epoch graphs for both methods

Task 4-
Use your model to get the sentiment predictions on this dataset (use the column **post_text** as input) and save them as a CSV file

Checkpoint-3: BERT

Welcome to the last checkpoint of this project!!
Hope you all really enjoyed this project and learned something
This week, we'll explore the BERT model(a SoTA Transformer)—its architecture, how it's trained, and how we can fine-tune it for our use-case.

Articles: Hugging Face's Blog, Jay Alammar's Illustration
Code: Fine-tuning, Text-Classification

Assignment-3:

Finetune the Bert-Uncased Model with our dataset (same as week-1,2). Use test_size=0.3 for splitting the data.
Then use your model to get the sentiment predictions on this dataset (use the column **post_text** as input) and save them as a CSV file

Note:

Training this transformer can be computationally expensive as it has around 110M parameters, so even using Colab's GPU may take 24 hours for this task.
So if you have good GPU access, then you can run it from your end, else you can send me the notebook once you are sure that only the training part's execution is left. I will share the notebook with you after running it on the GPU which I can access. Also please use PyTorch instead of Tensorflow if you intend to share the code with me.